2006

Animation in artificial grammar learning: can animation facilitate learning?

Bill Sallas
Louisiana State University and Agricultural and Mechanical College, sallas@lsu.edu

ANIMATION IN ARTIFICIAL GRAMMAR LEARNING: CAN ANIMATION FACILITATE LEARNING?

A Thesis

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Master of Arts

In

The Department of Psychology

by
Bill Sallas
B.S., University of Illinois, Urbana-Champaign, 1998
M.Ed., University of Illinois, Urbana-Champaign, 2001
May 2006

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Domangue, Mathews, Sun, Roussel, and Guidry (2004) trained participants to generate valid exemplars from an artificial grammar using either memory-based or model-based processing. Their results showed that learning by memory-based processing resulted in fast but inaccurate performance, while model-based learning resulted in slow but accurate performance. Attempts to integrate both types of training did not result in fast and accurate string generation. Fast and accurate performance was achieved by Sun and Mathews (2004) using a computer animated display to train participants. The current study used a 2x2x2 factorial design to determine why participants who view an animated display of a diagram of the grammar perform well at test. The results suggest that the diagram informs participants of which letters, or chunks of letters can appear in each position, as well as where they cannot appear. Animating the diagram focuses attention on the relevant portion of the complex display and leads to the best performance.

INTRODUCTION

Humans are capable of learning through two systems (Matthews, Buss, Stanley, Blanchard-Fields, Cho, and Druhan, 1989). Explicit learning is the conscious and effortful acquisition of rules (Reber 1993). An example is learning to solve an algebra problem by following a set of steps. Implicit learning is the non-conscious, automatic acquisition of information (Reber 1969). Infants learning to produce novel utterances without an explicit understanding of the rules used to generate those utterances is one example of implicit learning (Dienes, Broadbent, & Berry, 1991).

Much of the work studying explicit and implicit learning has used the artificial grammar learning paradigm, developed by George Miller (1958). In this paradigm a set of letter strings are generated using a set of rules that govern the placement of letters and the length of strings. Participants in Miller's original study were asked to generate letter strings without prior exposure to the grammar (rules) or exemplars, and an experimenter would provide feedback after each string had been generated. Participants rarely generated a correct string and became frustrated as the experimenter repeatedly gave negative feedback.

Reber (1969) adopted Miller's paradigm to study implicit learning. Unlike Miller, Reber exposed participants to a subset of the strings generated by the grammar in a study phase. To ensure implicit learning, participants were misled to believe that at some later point there would be a memory test for these strings. At test, it was revealed that the letter strings followed a set of rules and the participants were asked to discriminate between grammatical and ungrammatical stings. Reber (1969) chose to use a grammaticality judgment test rather than Miller's (1958) original string generation task so that learning could occur within one or two sessions in the lab. After exposure to valid strings, participants' performance on a grammaticality judgment test was above chance. Participants could not verbalize how they were performing the task, which led Reber to conclude that the knowledge they had learned about the grammar had been gained through an implicit system.

Since Reber's original work, some researchers have questioned whether learning can ever truly be implicit. Shanks and Channon (2002) argue that implicit learning tasks are not automatic, or nonconscious. Shanks and Channon used sequence learning, a typical implicit learning laboratory paradigm. Some participants were exposed to just the sequence learning task while others alsoperformed a secondary tone counting task. The authors argued that if sequence learning was automatic, the secondary task should not affect learning. This was not the case, as participants in the single-task condition performed significantly better at test. Shanks and Channon argue that because traditionally labeled implicit tasks can be affected by a secondary task, they are not nonconscious and therefore not implicit. The authors argue instead that learning occurs in one unitary explicit system.

Matthew et al. (1989) have suggested that the term implicit focuses too much on the nonconscious aspect of implicit learning. Instead they propose that humans learn through two separate systems; memory-based and model-based processing. During exposure to exemplars memory-based processing automatically abstracts patterns of covariance. Model-based processing is an explicit representation of the task which can guide actions within the task.

Mathews et al. (1989) also argued that memory-based and model-based processing can interact synergistically. Participants who first viewed exemplars from a bi-conditional artificial grammar (memory-based processing) and then corrected letter strings which contained errors (model-based processing), performed better on a grammaticality judgment test than participants who received the training in the opposite order or who received only one type of training. Additionally, when a finite-state grammar was used, where the rules are more difficult to generate than the relatively simple logical rules of a bi-conditional grammar, no synergy between memory-based and model-based processing was found. Therefore, when the rules were relatively easy for participants to generate, exposure to many exemplars (memory-based processing) followed by a task which encouraged model-based processing resulted in a synergy between the two types of processing. When the rules were difficult to generate, as in the finite-state grammar, this synergy did not exist.

Mathews et al. (1989) measured performance using a grammaticality judgment test. This method, developed by Reber's (1969), has been criticized for being too artificial by Mathews & Cochran (1998), who developed a cued-generate task as a more ecologically valid test of knowledge acquisition. While the grammaticality judgment test is not ecologically valid, it allows the researcher to test participants' knowledge of the grammar in a very precise manner. For example, the researcher could test the hypothesis that participants learn the beginning chunks of exemplars better than those that come in the middle by comparing error rates on items in which the error was in the beginning or middle of the letter string.

Unlike the Miller (1958) string generation task, where participants blindly combined letters in the hope of generating a valid string, the Mathews and Cochran's (1998) cued-generate task provides cues for the participant. A computer displays a set of dashes, corresponding to the number of letters in the target string, with two letters (the cues) from the target string displayed on two of the dashes. Participants fill in each blank dash from left to right with a letter and press the enter key. The participant's input is compared to the set of not-yet-generated strings from the grammar that have the same number of letters as the target string, and in which the cue letters appear in the same locations. Letters that match a valid string remain, while incorrect letters are erased and the participant is given another opportunity to fill in the blank dashes. When 70% of the dashes are filled with letters that match a yet-to-be-generated string, the computer displays that valid string. Mathews and Cochran (1998) described this as "computerized motherese". When a young child is learning to use language, the mother does not require perfection. Rather she understands what the child is saying and repeats the correct version back to the child.

Domangue, Mathews, Sun, Roussel, & Guidry (2004) used Mathews and Cochran's design to study the speed and accuracy with which participants could generate letter strings. In their memory-based processing condition, during training participants viewed valid strings and copied them with pen and paper. In the model-based processing condition, participants were given a diagram of the grammar, which was a visual representation of the rules used to generate the letter strings (see Figure 1). During training, participants viewed valid strings and wrote each letter from the string in its appropriate state, or position, in the diagram. Domangue et al. (2004) found that participants in the memory-based processing condition responded quickly but inaccurately on the cued-generate task. Conversely, those in the model-based processing training condition were slow but accurate on the task.

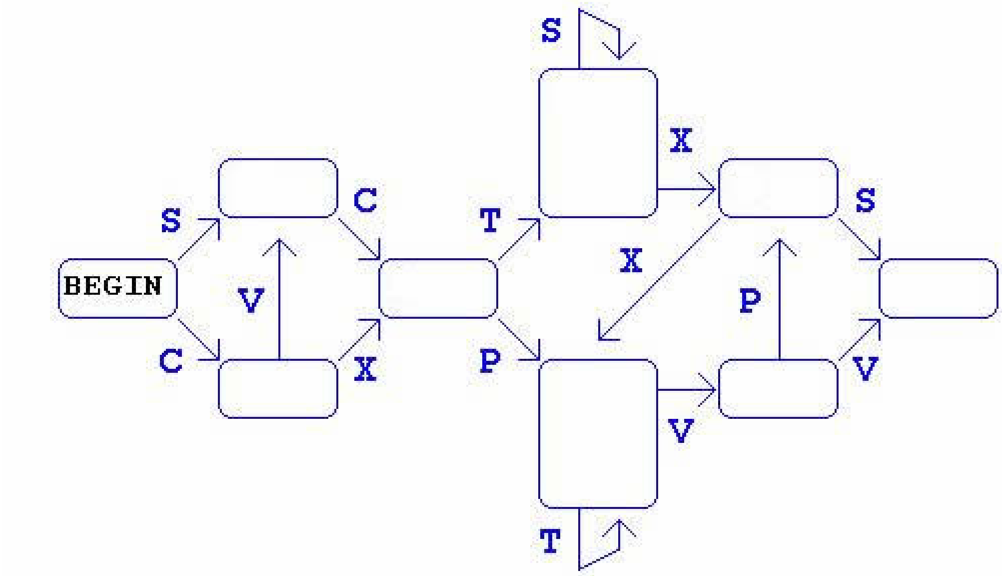

Figure 1. Diagram of grammar used in Domangue et al. (2004).

Domangue et al. (2004) also addressed synergy between memory-based and model-based learning. The authors attempted to make participants fast and accurate by integrating memory and model-based processing. However, this manipulation was unsuccessful.

Unlike Domangue et al. (2004), Sun and Mathews (2004) demonstrated that participants could achieve fast and accurate performance on the cued-generate task. Sun and Mathews compared performance on a transfer task involving string generation following training. Training was conducted through the use of three different computer games in which participants performed a string-edit task. The goal of all three training games was the same: participants were shown a letter string and told to identify the incorrect letters in that string. Their "score" was presented in terms of misses (incorrect letters that they did not identify as such) and false alarms (correct letters identified as incorrect). Participants were encouraged to respond quickly but not to sacrifice accuracy for speed and a monetary prize was offered to the participant in each condition who made the fewest errors. While the goal of the games was the same, they did differ in the type of assist cues given to the participant.

Participants in the letter-appearance condition attempted to identify the incorrect letters in the string without any assistance. They were shown a letter string at the bottom of the computer screen and told to select the incorrect letters in that string and click on them with the mouse. As the trial progressed, the computer presented the correct string at the top of the screen, with each letter appearing one-by-one from left to right, until the entire string was revealed. Approximately 3 seconds into the trial, letters began appearing at the top of the screen, and 500 ms before a letter appeared in its position at the top of the screen, participants could no longer edit the letter in that position, requiring fairly quick decisions.

In the primed-assist condition, participants were given the same string-edit task as the letter-appearance condition, but were provided an aid to prime correct choices. Instead of the correct letters appearing one-by-one at the end of the trial as in the letter-appearance condition, the letters began the trial at the bottom of the screen in an unreadable “bunch”.   The letters became recognizable as they spread out and floated from the bottom of the screen to their correct position at the top of the screen. A line was drawn across the middle of the screen. After the letters passed this line, participants could no longer mark the corresponding letters in the to-be-edited string as incorrect. As in the letter-appearance condition, participants were forced to make quick decisions.

Participants assigned to the diagram-assist condition were charged with the same string-edit task as the other conditions, but were provided with a diagram of the grammar for assistance. Instead of the letters floating to the top of the screen as in the primed-assist condition, the letters appeared, one-by-one, in the correct order and position in the diagram (see Figure 1 for example). Also, like the other two conditions, quick decisions were required; after a letter appeared in the diagram, participants could no longer click on the corresponding letter in the string.

Following training, a transfer test using the same cued-generate task as in Mathews and Cochran (1998) and Domangue et al. (2004) was used to compare performance across conditions. The results from that transfer task showed that participants in the diagram-assist condition responded as quickly as those in other conditions, but were more accurate than the other conditions in their responses (Sun & Mathews, 2004).

This is an interesting finding because the diagram-assist group in Sun and Mathews (2004) received the same information as the participants who traced exemplars through the diagram in Domangue et al. (2004). Both groups viewed valid exemplars in the context of a diagram of the grammar. The main difference was that participants in the Domangue et al. study manually copied letters from exemplars into the diagram themselves, while those in the Sun and Mathews study viewed an animation of the letters appearing in the diagram and used that animation to complete the string-edit task.

This difference in performance may have been due, in part, to the way participants viewed the exemplars. The exemplar diagramming task in Domangue et al. encouraged the parsing of whole exemplars into individual letters and placing those letters in the diagram. While this type of training provided knowledge of how exemplars are constructed, it likely failed to focus attention on encoding exemplars. In Sun and Mathews, the animated diagram condition was designed to provide insight into how the exemplars are formed and prime an integrated encoding of the exemplar, thus creating a synergy between model-based and memory based processing.

It is likely that the exemplars in Sun and Mathews were not encoded as a whole, but rather in two and three letter chunks. Servan-Schreiber and Anderson (1990) compared performance on a grammaticality judgment task between participants who viewed whole exemplars during training and participants who viewed the same exemplars segmented into chunks of two and three letters. Performance of both groups was similar, leading the authors to conclude that participants in the whole exemplar condition divided the letter strings into chunks, like had been done by the experimenter for the other condition. They also contend that learning in artificial grammar tasks is a process of segmenting strings into an increasingly well organized hierarchical network of chunks. The more efficient the chunking system, the more likely strings will be correctly classified on a grammaticality judgment task. While their hierarchical model does not incorporate chunk order, it is clear that order does affect ability to make grammatical judgments. Servan-Schreiber and Anderson noted that strings with errors at the end are more likely to be judged ungrammatical than those with errors at the beginning or in the middle.

This idea of chunking has also been endorsed by Perruchet and Pacteau (1990). Perruchet and Pacteau trained participants by either exposing them to a set of exemplars or to just the bigrams and trigrams that made up those exemplars. On a later grammaticality judgment test, participants who saw only the bigrams and trigrams performed as well as participants who saw whole exemplars. It is important to note that the grammar used in this study did not require participants to have knowledge of the correct dependencies of chunks within the string. Like Servan-Schreiber and Anderson (1990), Perruchet and Pacteau argued that only knowledge of chunks is necessary for better-than-chance performance on grammaticality judgment tests.

In Sun and Mathews (2004), it seems reasonable that subjects were chunking the exemplars, and using those chunks to generate strings in the cued-generate task. In fact, when asked during exit interviews, many participants reported that they knew that the letters "CVC", "VPS", and other two and three letter chunks often appeared together. Additionally, it is unlikely that participants in the animated diagram condition were literally using a mental model of the diagram to guide their responses for two reasons. First, they responded as quickly as participants in the other conditions. One would expect that using a mental representation of the diagram would be slow (Norman, 1993). Second, when asked, most participants did not report trying to remember the diagram or using it on the cued-generate task. Their responses during the exit interview were similar to those of participants in other conditions, reporting that they remembered chunks of letters that often appeared together.

There are at least three possibilities to explain the fast and accurate generation performance of participants in the diagram-assist condition in Sun and Mathews (2004). First, the diagram may have provided inter-letter dependency information that the other conditions did not explicitly receive. For example, the diagram shows that the letter "C" can only appear within the first three positions of a valid exemplar (see Figure 1). Rather than only remembering chunks, the diagram may help participants remember the chunks as well as the correct dependencies of the chunks. This would explain why the diagram-assist condition is able to respond more accurately than the other conditions.

A second possibility is that the animated diagram task may have encouraged deeper processing than the other training tasks (Craik & Lockhart, 1972). When participants saw a letter appear in the diagram, they were forced to make a decision about what the next letter in the string could be and then compare their prediction to the to-be-edited string. Making predictions about the next letter may have resulted in deeper processing compared to the primed-assist group in which the participants could passively watch the letters rise to the top of the screen and make comparisons between the rising letters and the letters in the to-be-edited string.

The third possibility for the fast and accurate performance in Sun and Mathews (2004) is that their training task was animated, while the training task in Domangue et al. (2004) was a static pen and paper test. While it is impossible to compare performance across studies, it does seem possible that animating the diagram in Sun and Mathews may have had a positive effect on participants' learning by providing information as it is needed to encode the exemplars. Animation may add a temporal element that is lacking in a static display. By displaying exemplars over time, animation may help participants to encode dependencies between each letter or chunk. While this information is certainly available in the static display, it may become more salient when animated.

While there has been some research showing the facilitative effects of animation, Tversky, Morrison, and Betrancourt (2002) claimed that this research does not demonstrate that animation improves learning. They argued that these studies do not equilibrate the animated and static conditions. For example, Large, Beheshti, Breuleux, and Renaud (1996) used animated and static displays of the heart in a lesson on blood circulation and showed an advantage for learning in the animated condition. However, the animated graphics of the heart showed additional blood pathways which were not included in the static displays. Another problem with animation research, noted by Tversky et al. (2002), is the difference in interactivity between animated and static displays. In Schnotz and Grzondziel (1999), participants in the animated condition had the option of interacting with the display, while those in the static condition could only look at the static graphic.

Even when these confounds are not present, researchers have a difficult time showing a benefit of animation. Rieber, Boyce, and Assad (1990) used static and animated displays to teach Newton's laws to college students and found no effect of display type. Similarly, when teaching peptide chain formation, ChanLin (1998) found no advantage for an animated display over the static graphic. Finally, Palmiter, Elkerton, and Baggett (1991) found that retention after a one week delay was worse for students using animated display than those who viewed a static display.

Tversky et al. (2002) argued that animations may not provide a beneficial effect for two reasons. First, motion in animations may be difficult to perceive. Just as in the real world, movement, trajectory, and the interaction of moving parts in animations can be difficult for people to perceive. For example, Caramazza, McCloskey, and Green (1983) reported that people rely on inaccurate perceptions rather than the laws of physics when perceiving motion. Second, Tversky et al. (2002) claimed that people often perceive complex motion as discrete steps which are better displayed in static images that can be reviewed and compared step-by-step. Animations change and when finished, cannot be reviewed like a static diagram.

The current study addressed the problems in research on animation described by Tversky et al. (2002). Unlike some of the studies mentioned above, the same information was available in both static and animated displays. Additionally, the stimuli in the current study were not derived from motion-intensive material (e.g. blood circulation, Newtonian physics) which may have created perceptual difficulties in previous studies.

The current study attempted to tease apart which of the three components of the Sun and Mathews (2004) animated diagram training task facilitated fast and accurate performance at test using a 2x2x2 factorial design, with display type (animated and static), content (diagram and chunk), and prediction (immediate or predictive) as factors. As in Sun and Mathews (2004), during training participants performed a string-edit task in which they identified incorrect letters with various assist cues to help in this task. In the training task, participants saw the assist cues either in chunks or in the context of the diagram. Also, the cues were either static or animated. Finally, the cues were either available immediately or became available only after participants had edited the string (predictive). These factors combined to form eight training conditions: diagram animated predictive, diagram static predictive, diagram animated immediate, diagram static immediate, chunk animated predictive, chunk static predictive, chunk animated immediate, and chunk static immediate. A no training control group was also run. In addition to the cued-generate test, a grammaticality judgment test and a recognition test were also administered.

If the animated diagram (Sun & Mathews, 2004) only provided additional information about the inter-chunk dependencies, participants in the current study who view the diagram should be only as accurate as those who view the exemplars parsed into chunks at training. Like the diagram, segmenting the exemplars into chunks makes the inter-chunk dependencies more salient relative to seeing the exemplar as a whole.

The advantage of viewing the animated diagram in Sun and Mathews may have been that it encouraged deeper processing because participants were required to make predictions about which letter would appear next in the diagram rather than making one-to-one comparisons. If that were the case, participants who make predictions in the current study should be more accurate and faster on the cued-generation task than participants who do not make predictions.

The temporal element added by animation in Sun and Mathews (2004) may have made the dependencies between letters more salient. If this were the case, participants in the current study who view an animated display should be more accurate on the cued-generation test than those who view a static display.

In the eyewitness literature, well-developed scripts can have a negative impact on memory for specific events (Greenberg, Westcott, & Bailey, 1998). In the same way, as knowledge of how chunks can be combined to generate strings increases, memory for specific instances may be decreased (Mathews, 1991). If participants who view the diagram gain insight into how the letter strings are formed, this general model-based memory could hurt memory for specific instances seen during training. Participants were given a recognition test for items seen at training to test this hypothesis.

A grammaticality judgment test (Reber, 1969) was also given to test if the utility of an animated diagram (Sun & Mathew 2004) was due to the additional dependencyl information it provides or because the task encourages deeper processing. In Anderson's (1976) ACT model, strong connections in a network yielded faster responses than weak connections. If the animated diagram provided only additional dependency information, there should be no between group differences in response latency on this test. However, if predicting which letters will appear next encourages deeper processing, one would expect to see faster response times in the predictive groups due to stronger connections within their networks.

To gain further insight into how participants organize information in their networks of chunks (Servan-Schreiber & Anderson 1990), the grammaticality judgment test included several error types. Some of the errors were due to a valid chunk, or group of letters that can legally appear in order, in the wrong position within the letter string (between chunk errors). Other incorrect strings were created by substituting one letter for another within a valid chunk (within chunk errors). If participants who were trained using the diagram received additional inter-letter dependency information, they should reject letter strings that violate order (position) more than participants who are in the chunk conditions. No group differences were expected in the within chunk errors.

METHOD

Participants

One-hundred and eighty-seven participants were recruited for this study. Participants were drawn from the subject pool of psychology students taking a variety of courses at Louisiana State University. All participants were volunteers and received extra credit for their participation.

Materials

The finite-state artificial grammar used by Domangue et al. (2004) and Sun and Mathews (2004) was used in the current study. The grammar generates 177 letter strings using the letters, S, C, V, X, T, and P. The letter strings in the grammar range in length from five to eleven letters.

Design

This study used a 2x2x2 factorial design, with content (diagram or chunks), presentation type (static or animated), and prediction (predictive or immediate) as the between-subject factors. A test-only control was also run so that comparisons could be made between participants who were trained with various assist-cues and those whose training consisted solely of viewing exemplars in the context of the cued-generation test.

Procedure

Participants were tested in groups up to 8. Participants completed five 1-hour sessions over the course of one week. Sessions one through four consisted of a 20-minute study phase and 20-minute string-generation test, with the test-only control condition completing only the string-generation test at each session. In session five, participants completed a 20-minute string-generation test, followed by a grammaticality judgment test, and an episodic memory test for exemplars seen in training.

Training Phase

Training was conducted through the use of a computer game in which participants performed a string-edit task (Sun & Mathews, 2004). Participants were shown a letter string and instructed to identify the incorrect letters in that string by clicking on those letters with a mouse. Their score was presented in terms of misses (incorrect letters that they did not identify as such) and false alarms (correct letters identified as incorrect). Participants were encouraged to respond quickly but not sacrifice accuracy for speed. A monetary prize was offered to the participant who made the fewest errors to further emphasize accuracy over speed. Each time a letter string was displayed, it contained between one and four errors randomly generated by the computer at the beginning of each trial. Participants read a short description of the cover story, which explained that they were learning to make corrections in secret code words.

Each training session was 20-minutes in length and consisted of approximately 88 trials. To determine which exemplars would be shown during training, a subset of eleven exemplars from the corpus of 155 was randomly selected for each participant by the computer at the beginning of each session. The set is drawn from 155 exemplars because twenty-two of the 177 exemplars were withheld for later use as lures in the episodic memory test. An additional eleven exemplars were drawn from the representative set of 22 exemplars used in Domangue et al. (2004) (see appendix A). This set was divided into two subsets of 11 exemplars each, sets A and B. During each training session, one of the sets was added to the 11 exemplars selected by the computer for a total training set of 22 exemplars. The sets from Domangue et al. were rotated so that sessions one and three used set A and sessions two and four used set B. The exemplars from these repeated sets were used in the episodic memory test in session five.

The order in which the exemplars were displayed in the string-edit task was determined by the computer in a random selection without replace procedure. When each exemplar had been displayed once, the computer generated a new order. Participants viewed each exemplar approximately four times during the study phase.

Like Sun and Mathews (2004), participants were given assistance cues to complete the string-edit task. In the static diagram immediate condition participants saw a diagram of the grammar used to generate the exemplars. At the beginning of each trial, all of the letters in the exemplar were shown in their appropriate state in the diagram. Participants could then compare the letters in the diagram to the letters in the string they were editing and mark errors where appropriate. Three seconds after the trial began, a dot appeared under the first letter in the to-be-edited string. After 500ms, the dot moved to the next letter in the to-be-edited string and the participant was no longer allowed to mark the first letter as incorrect. After another 500 ms interval, the dot would move to the third letter in the string and the participant was no longer allowed to edit the second letter. Thus, the dot was a visual timing device which forced participants to make quick decisions in the edit task. After the dot passed a letter, an unmarked error was recorded as a miss, and an “X” mark appeared over that letter, alerting participants of their error. False alarms were also marked with an “X”. The timing dot and “X” marks for errors were consistent across all condition. At the end of each trial, the exemplar was displayed at the top of the screen. In addition, a feedback screen was displayed which informed participants of how many misses and false alarms they had made on the previous trial.

In the static diagram predictive condition, participants performed the string-edit task while the diagram was displayed on the screen. At the end of each trial, all of the letters from the exemplar appeared simultaneously in their appropriate state within the diagram (see Figure 2 for an example of the diagram training task).

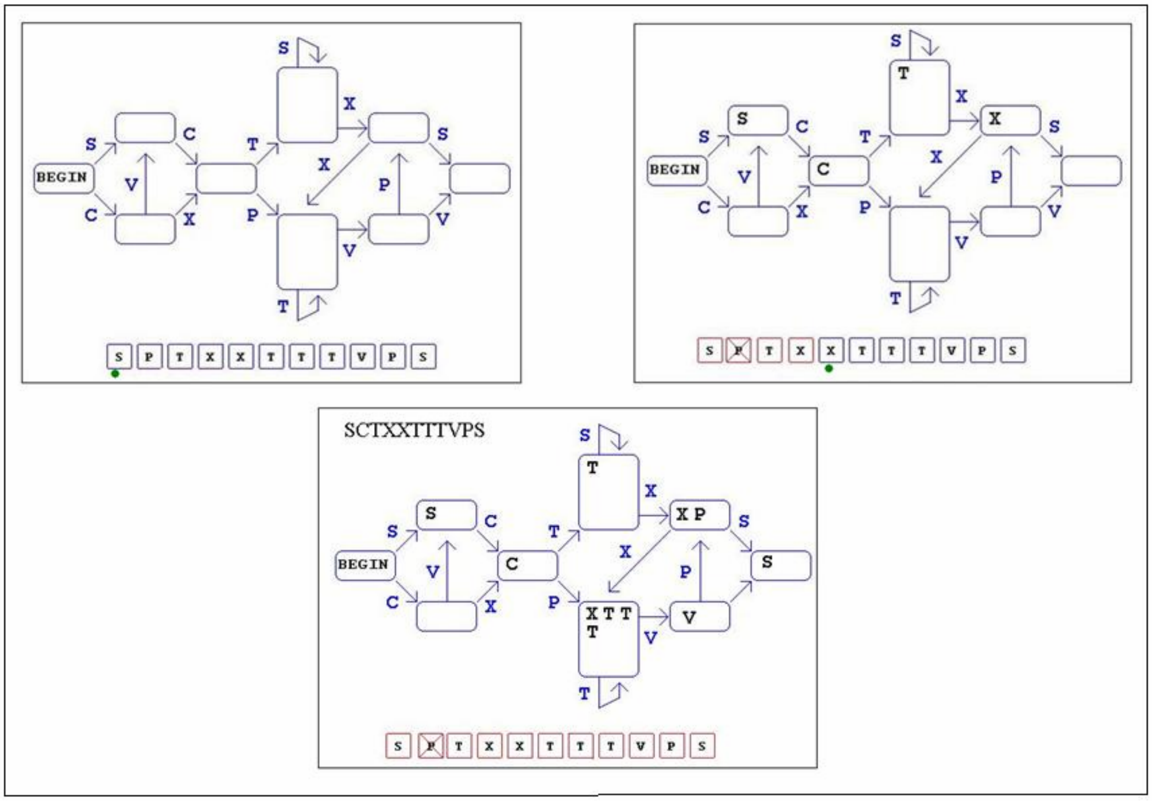

Figure 2. Three slides showing the progression within the animated diagram predictive training task.

Participants in the animated diagram immediate condition saw the letters appear one-by-one in the diagram. After a letter appeared in the diagram, participants had 500 ms to compare that letter to the corresponding letter in the string they were editing and mark an error if necessary. No predictions needed to be made as participants were able to directly compare letters in the to-be-edited sting to the correct letters in the diagram.

Participants in the animated diagram predictive condition also saw the letters appear one-by-one in the diagram. However, they were forced to make predictions about which letter will appear next in the diagram because after a letter appeared in the diagram, participants were not able to mark the corresponding letter in the to-be-edited string as an error. The letters appeared in the diagram at 500 ms intervals.

The remaining four conditions performed the same string-edit task. In the chunk conditions, the letters from the exemplar appeared in the center of the screen, from left to right, in chunks of two or three letters rather than in a diagram of the grammar. Space was left between the chunks to make them more salient.

To divide the exemplars into chunks 25 undergraduates were recruited for a pilot study from the Louisiana State University psychology subject pool. Participants had no prior experience with artificial grammar learning experiments or with the particular grammar used. Participants were given a packet of the 177 exemplars of the grammar with approximately 30 exemplars printed on each page. The exemplars were arranged in a random order. Participants were told to read the letters in the exemplars from right to left, and place a line between letters where they naturally paused. Participants completed the first packet and then were given a second packet. The second packet contained all 177 exemplars in a different order from the first packet. Participants were given the same instructions. Perruchet, Vinter, Pacteau, and Gallego (2002) found that segmentation of exemplars became consistent across participants with increased exposure, so only data from the second packet were used.

Each division participants made was considered a chunk, and the frequency of each chunk was recorded. The goal was to develop the smallest set of chunks which could, in various combinations, produce any exemplar in the corpus. With this in mind, the most frequently generated chunks from the pilot study were identified. From that list, the smallest set (17) of chunks that could generate the entire corpus of exemplars was selected (see appendix B).

Participants in the static chunk immediate condition saw the entire exemplar, segmented into chunks, at the beginning of each trial. Like the static diagram immediate condition, participants were able to compare letters in the string they were editing to the letters that appeared in the chunked exemplar. Also, like all other conditions, a dot appeared under each letter in the to-be-edited string to alert participants that their time to mark an error in the letter position was almost over (see Figure 3 for an example of the chunk training task).

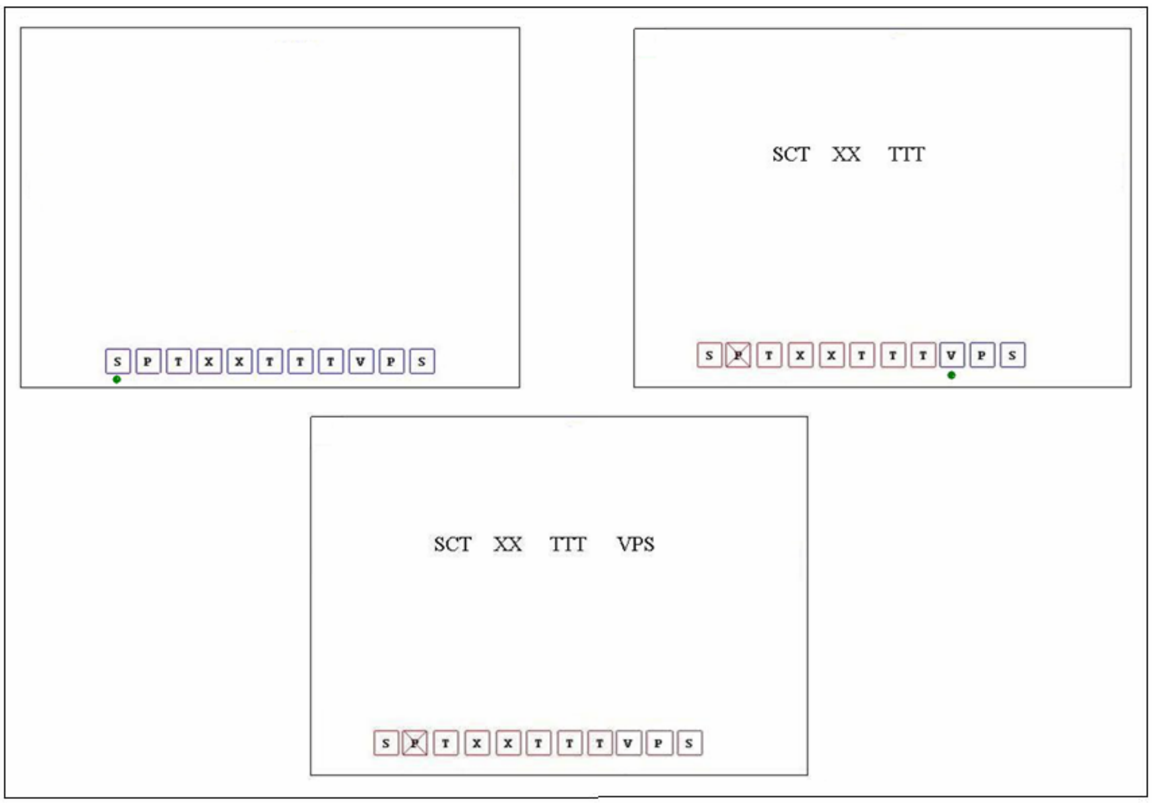

Figure 3. Three slides showing the progression within the animated chunk predictive training task.

Participants in the static chunk predictive condition saw only the to-be-edited string at the beginning of each trial. At the end of the trial the chunked letter string appeared.

In the animated chunk immediate condition, each chunk appeared one at a time, from left to right. As the chunks appeared on the screen, participants were able to compare letters in the to-be-edited string to those in the exemplar and mark any incorrect letters. After a chunk appeared, participants had 500 ms per letter in the chunk to mark an error in the corresponding letters in the to-be-edited string. Like all other conditions, participants were made aware of the timing by the dot that moved from letter to letter.

Finally, in the animated chunk predictive condition, the chunks appeared one at a time. If the first chunk in the exemplar was three letters in length, the timing dot would move through the first three letters in order, stopping for 500 ms at each letter. When the dot moved to the fourth letter, the first chunk would appear. Thus, the participants were forced to predict what letters would appear in the first chunk before seeing the correct chunk of letters appear. Immediate feedback was given if participants marked a correct letter in the to-be-edited string as correct, or if an incorrect letter was not marked after the timing dot had passed.

At the beginning of each training phase, participants completed three practice trials. These practice trials were similar to the actual trials except that they use vowels instead of the consonants used in the grammar. During the practice trials, the experimenter answered any questions the participants may have had.

Cued-generation Test

After the 20-minute training phase, participants were given a short break and then read the instructions for the test phase. Participants were instructed that their task was to find as many secret code words (exemplars) as possible in the 20-minute test period. They were also told that they should work as quickly and as accurately as possible and the participant who found the most code words would receive a monetary prize.

Participants in the no-training control were given a sheet of paper with the letters from the grammar printed horizontally across the page in a random order. This was done because they had no prior experience with the grammar and did not know which letters were available for use.

At the beginning of each test trial, the computer randomly selected a target exemplar. A set of dashes were displayed on the screen, with one dash for each letter in the target. Two letters, the cues, were displayed in their correct position above the appropriate dashes. Working from left to right, participants typed letters, one for each dash. When they came to a letter that was already filled in, they retyped that letter. When all the dashes were filled in, the participant pressed the enter key. The computer then compared the letters that the participant entered with all of the not-yet-generated exemplars in the database. If the string typed by the participant did not match at least 70% of the letters in a valid exemplar, the computer erased any incorrect letters, leaving only letters that matched the closest valid exemplar left in the database. Participants continued this process until at least 70% of the letters matched a not-yet-generated-exemplar. Exemplars have pairs of letters in common so it was not necessary for the participant to type the target exemplar chosen by the computer. When the participant reached the 70% criterion, a feedback screen appeared in which the letter string generated by the participant was displayed. The computer also displayed the closest matching exemplar, and the percentage of letters from the participant-generated string which matched that exemplar. Once one of the 155 exemplars had been produced, it was removed from the database and could not be generated again until the next test session. The database included only 155 exemplars because the same 22 exemplars that were withheld from the study phase for later use as lures in the recognition test were withheld here as well.

During the first four sessions, participants completed the training phase followed by the test phase. There was no training phase in the fifth session. The 20-minute cued-generation test phase was administered, followed by two additional tests: grammaticality judgment and episodic memory tests.

Grammaticality Judgment Test

After the cued-generation test, participants completed a grammaticality judgment test. At this point, the participants were instructed that the letter strings that they had seen during the past four sessions followed a set of rules (Reber 1969). They were told that they would see letter strings on the computer screen and they should press one key if the letter string followed those rules and another key if the letter string did not follow the rules. Participants were told that they should make their decisions as quickly as possible but not sacrifice accuracy. Response latency and accuracy measures were recorded.

The grammaticality judgment test consisted of 100 valid exemplars and 140 invalid lures, which could be divided into two groups. One type of lure was created by substituting one intact chunk for another. In some cases, the new chunk came from the same position in the exemplar (i.e. substituting one beginning chunk with another that could not be followed by the rest of the exemplar). In other cases a chunk was replaced with a chunk from a different location (i.e. a beginning chunk replaced by a chunk from the end of an exemplar.) The second type of error was created by changing one or all of the letters in a chunk to make it invalid (see appendix C for a complete list of invalid chunks).

Episodic Memory Test

After completing the grammaticality judgment test, participants were given a short break and then completed an episodic memory test where they were asked to identify letter strings that they had seen during the training phases. The targets on the episodic memory test were the twenty-two repeated exemplars (sets A and B) from training. Each target was seen approximately 8 times during training. An equal number lures were randomly drawn from the corpus of 177 exemplars. The lures were the same across all participants. These twenty-two lures were removed from the corpus, so they were never seen in the training or generation phases.

Participants were instructed that they would see a letter string on the computer screen, and that all of the letter strings would be real code words (valid exemplars). They were to press one key if they had seen the letter string before and another if they had not seen the letter string before. Participants were instructed to respond as quickly as possible while still being accurate. Feedback was given after each trial. The number of correct responses was recorded.

RESULTS

Speed

Speed was a measure of the total number of attempts made per minute during the 20-minute cued-generation test. The means for the speed data are presented in Table 1. A three-way between-subjects analysis of variance (ANOVA) was run on the speed scores. There was a trend towards a main effect of content, $F(1, 158) = 3.75$, $p = .055$. While the effect did not reach significance, participants who viewed the diagram in training made more attempts at test ($M = 8.56$) than those who viewed chunks ($M = 7.71$).

The ANOVA also revealed a significant interaction between prediction and display type, $F(1, 158) = 4.85$, $p < .05$. However, follow up tests of simple effects failed to reveal any significant differences between the means.

Table 1. Mean scores on the speed measure

		Predictive	Immediate
Chunk			
	Static	6.94 (.58)	8.23 (.58)
	Animated	7.92 (.63)	7.69 (.61)
Diagram			
	Static	8.2 (.58)	8.80 (.60)
	Animated	9.37 (.60)	7.81 (.63)
Test-Only Control: 9.2 (.56)			

Note. Standard deviations presented in parentheses.

A one-way ANOVA (including the test-only control) was not significant by group F(8, 177)=1.77, *ns*.

Accuracy

Accuracy was a measure of the proportion of letter strings generated on the first attempt that matched 100% of the letters in the target exemplar per minute. The means for the accuracy data are presented in Table 2. A three-way between-subjects ANOVA was run on the accuracy data from the cued-generation test. There was a significant main effect of display content, $F(1, 158) = 4.78$, $p < .05$. Participants who saw the diagram at training produced a greater number of perfect exemplars on the first attempt ($M = .616$) than participants who saw chunks at training ($M = .366$).

The ANOVA also revealed a significant interaction between display type and content, $F(1, 158) = 4.23$, $p < .05$. Follow up tests of simple effects revealed that participants who viewed the animated diagram generated more perfect strings per minute ($M = .84$) than those who viewed the animated chunk display ($M = .34$), $F(1, 77) = 6.5$, $p < .05$. When the display was static, participants who viewed the diagram ($M = .41$) did not generate significantly more perfect strings than participants who viewed chunks ($M = .39$), $F(1, 85) = .016$, *ns*.

Table 2. Mean scores on accuracy measure.

		Predictive	Immediate
Chunk			
	Static	.45 (.51)	.33 (.31)
	Animated	.38 (.39)	.31 (.32)
Diagram			
	Static	.46 (.83)	.53 (.65)
	Animated	1.04 (1.4)	.62 (.87)
Test-Only Control: .01 (.044)			

Note. Standard deviations presented in parentheses.

A one-way ANOVA (including the test-only control) was significant by group F(8, 177)=13.31, p < .01. A Dunnett's t-test compared all groups against the control and found the diagram animated immediate, diagram animated predictive, and diagram static immediate groups were more accurate than the test-only control (see Table 3 for the mean difference between each group and the test only control.)

Table 3. Results of Dunnett t-tests for each measure.

Group	Accuracy	Achievement	Grammaticality RT	Grammaticality Accuracy (Overall)	Grammaticality Accuracy (chunk errors)	Grammaticality Accuracy (position errors)
Chunk Animated Immediate	.3	2.5**	726.54**	.11**	.14*	.03
Chunk Animated Predictive	.37	2.4**	958.05**	.15**	.21*	.003
Chunk Static Immediate	.32	2.3**	625.86**	.12**	.13*	.03
Chunk Static Predictive	.44	2.4**	752.67**	.12**	.16*	.02
Diagram Animated Immediate	.61*	2.4**	713.1**	.09*	.1	.05
Diagram Animated Predictive	1.0*	3.0**	613.9**	.12**	.1	.1*
Diagram Static Immediate	.34*	1.7*	361.56	.06	.05	.01
Diagram Static Predictive	.45	2.0*	496.73*	.09**	.09	.03

Note. Mean difference between test-only control and each group is shown. * $p < .05$, ** $p < .01$

Achievement

Achievement was a measure of the proportion of letter strings generated on the first attempt that matched at least 70% of the letters in the target exemplar per minute. The means for the achievement data are presented in Table 4. A three-way between-subjects ANOVA was run on the accuracy data from the cued-generation test. No significant effects were found.

Table 4. Mean scores on achievement measure.

		Predictive	Immediate
Chunk			
	Static	2.62 (1.49)	2.6 (1.56)
	Animated	2.69 (1.91)	2.46 (1.56)
Diagram			
	Static	2.57 (2.21)	1.98 (1.88)
	Animated	3.25 (3.32)	2.67 (2.25)
Test-Only Control: .26 (.47)			

Note. Standard deviations presented in parentheses.

A one-way ANOVA (including the test-only control) was significant by group $F(8, 177)=4.34$, $p < .001$. A Dunnett's t-test compared all groups against the control and showed that all groups had a higher level of achievement than the test-only control.

Grammaticality Judgment Reaction Time

A three-way, between-subjects ANOVA was run on reaction time data from the grammaticality judgment test. There was a significant main effect of display type, $F(1, 158) = 4.67$, $p < .05$. Participants who saw a static display at training ($M = 1654.81$ s) responded more quickly than participants who saw an animated display ($M = 1842.37$ s). There was also a significant main effect of content, $F(1, 158) = 6.0$, $p < .05$. Participants who saw a diagram at training ($M = 1635.03$ s) responded more quickly than participants who saw the exemplars parsed into chunks ($M = 1853.11$ s). There were no significant interactions. The means for the RT data are presented in Table 5.

Table 5. Mean reaction times on grammaticality judgment test.

		Predictive	Immediate
Chunk			
	Static	2051.4 (461.7)	1719.2 (421.4)
	Animated	2051.4 (461.7)	1819.9 (503.5)
Diagram			
	Static	1590 (611.3)	1454.9 (506.7)
	Animated	1707.2 (756.5)	1806.4 (724.3)
Test-Only Control: 1093.3 (497.1)			

Note. Standard deviations presented in parentheses.

A one-way ANOVA (including the test-only control) was significant by group $F(8, 177) = 4.68$, $p < .001$. A Dunnett's t-test compared all groups against the control and showed that all groups except the diagram static immediate responded more slowly than the test-only control.

In addition, two types of errors were examined individually. The first type of error was created by substituting one intact chunk for another. Table 6 presents mean reaction time data. When a chunk was in the wrong position, there was a significant main effect of content, $F(1, 158) = 8.9$, $p < .01$. Participants who saw a diagram at training (M

= 1603 s) responded more quickly than participants who saw the exemplars parsed into chunks (*M* = 1860 s). There was also a main effect of display type $F(1, 158) = 5.1$, $p < .05$. Participants who saw a static display at training (*M* = 1641 s) responded more quickly than participants who saw an animated display (*M* = 1831 s).

Table 6. Mean reaction times for interchunk error items.

		Predictive	Immediate
Chunk			
	Static	1837.3 (583.7)	1710.2 (427.4)
	Animated	2089.7 (485.9)	1832.8 (504.2)
Diagram			
	Static	1579.8 (608.3)	1427.9 (479.1)
	Animated	1622.8 (733.1)	1757.4 (652.4)
Test-Only Control: 1088.9 (466.8)			

Note. Standard deviations presented in parentheses.

The same pattern of results occurred when the error was within a chunk (see Table 7 for means). There was a significant main effect of content $F(1, 158) = 5.9$, $p = .05$. Participants who saw a diagram at training (*M* = 1583 s) responded more quickly than participants who saw the exemplars parsed into chunks (*M* = 1798 s). There was also a main effect of display type $F(1, 158) = 4.6$, $p < .05$. Participants who saw a static display at training (*M* = 1603 s) responded more quickly than participants who saw an animated display (*M* = 1787 s).

Table 7. Mean reaction times for intrachunk error items.

		Predictive	Immediate
Chunk			
	Static	1787.7 (537.4)	1684.1 (373.7)
	Animated	2062.6 (702.0)	1697.1 (447.0)
Diagram			
	Static	1523.6 (566.5)	1420.9 (473.0)
	Animated	1636.1 (690.0)	1776.0 (710.0)
Test-Only Control: 1107.2 (517.2)			

Note. Standard deviations presented in parentheses.

Grammaticality Judgment Accuracy

Overall grammaticality judgment accuracy was a proportion of the number of letter strings correctly classified as valid and invalid divided by the total number of items (see Table 8 for mean accuracy data). A three-way, between-subjects ANOVA was run on accuracy data from the grammaticality judgment test. There was a main effect of content, $F(1, 158) = 4.42$, $p < .05$. Participants who saw chunks in training ($M = .67$) were more accurate than those who saw the diagram at training ($M = .63$).

Table 8. Mean overall accuracy data from grammaticality judgment test.

		Predictive	Immediate
Chunk			
	Static	.66 (.073)	.66 (.065)
	Animated	.69 (.069)	.66 (.095)
Diagram			
	Static	.63 (.099)	.60 (.115)
	Animated	.67 (.151)	.63 (.108)
Test-Only Control: .54 (.054)			

Note. Standard deviations presented in parentheses.

A one-way ANOVA (including the test-only control) was significant by group $F(8, 177) = 4.28$, $p < .001$. A Dunnett's t-test compared all groups against the control and showed that all groups except the diagram static immediate responded more accurately than the test-only control.

When a chunk was in the wrong position, there was a significant main effect of content, $F(1, 158) = 10.89$, $p < .01$ (see table 9). Participants who saw a diagram at training ($M = .58$) were more accurate than participants who saw the exemplars parsed into chunks ($M = .52$).

There was also a significant main effect of prediction, $F(1, 158) = 4.34$, $p < .05$. Participants who made predictions at training ($M = .57$) were more accurate than participants who did not make predictions ($M = .53$).

While there were no significant interactions, there was a trend towards an interaction between content and display F(1, 158) = 3.32, $p = .07$. Participants who viewed the animated diagram were more accurate (M = .60) than those who viewed the static diagram (M = .55).

Table 9. Mean accuracy for interchunk error items.

		Predictive	Immediate
Chunk			
	Static	.55 (.023)	.5 (.023)
	Animated	.53 (.025)	.5 (.024)
Diagram			
	Static	.56 (.023)	.54 (.024)
	Animated	.63 (.024)	.58 (.025)
Test-Only Control: .53 (.024)			

Note. Standard deviations presented in parentheses.

A one-way ANOVA (including the test-only control) was significant by group $F(8, 177) = 2.75$, $p < .01$. A Dunnett's t-test compared all groups against the control and showed that only the diagram animated predictive group responded more accurately than the test-only control when one valid chunk was substituted for another.

When there was an error within a chunk, a different pattern of results emerged (see Table 10 for means). There was a significant main effect of content $F(1, 158) = 10.24$, $p = .01$. In this case, participants who saw the exemplars parsed into chunks at training ($M = .75$) were more accurate than participants who saw a diagram ($M = .67$). There were no significant interactions.

Table 10. Mean accuracy for intrachunk error items.

		Predictive	Immediate
Chunk			
	Static	.75 (.032)	.72 (.032)
	Animated	.80 (.034)	.73 (.033)
Diagram			
	Static	.68 (.032)	.64 (.032)
	Animated	.69 (.032)	.69 (.034)
Test-Only Control: .59 (.033)			

Note. Standard deviations presented in parentheses.

A one-way ANOVA (including the test-only control) was significant by group $F(8, 177) = 3.39$, $p < .01$. A Dunnett's t-test compared all groups against the control and showed that all groups that saw chunks during training performed more accurately than control, while those that saw the diagram at training did not differ from control.

Episodic Memory Test

Accuracy on the episodic memory test was a proportion of the number of exemplars correctly classified old and new divided by the total number of items. There were no significant main effects. Overall, participants performed at chance ($M = .496$) (see Table 11).

Table 11. Mean accuracy on episodic memory test .

		Predictive	Immediate
Chunk			
	Static	.49 (.08)	.49 (.07)
	Animated	.51 (.07)	.496 (.08)
Diagram			
	Static	.48 (.08)	.49 (.08)
	Animated	.52 (.07)	.49 (.07)

Note. Standard deviations presented in parentheses.

DISCUSSION

When learning abstract material, one approach involves exposure to many examples of the corpus (memory-based), while a second approach involves learning the underlying structure (model-based). Domangue et al. (2004) found memory-based processing leads to fast but relatively inaccurate performance, while model-based processing leads to slow but accurate performance at test. Their attempts to integrate memory and model-based training to facilitate fast and accurate performance were unsuccessful. However, Sun and Mathews (2004) were successful in facilitating fast and accurate performance at test using an animated training task.

The purpose of this study was to tease apart which of the three components of the Sun and Mathews' (2004) animated diagram training task facilitated fast and accurate performance at test. Three possibilities were explored. First, the diagram may have provided correct chunk dependency information that was not explicitly available in the other conditions. Second, participants in Sun and Mathews who viewed the diagram were forced to make predictions about which letter would appear next. Making predictions rather than having the information immediately available may have encouraged deeper processing (Craik & Lockhart, 1972). Finally, animating the diagram may have made the ordinal position of all letters more salient than in the other conditions.

The first hypothesis was that the animated diagram only provided correct chunk dependency information that was not explicitly available in other conditions. If this hypothesis were true participants who viewed the exemplars parsed into chunks should have been as accurate on the cued-generation test as those who view the diagram in training. This is because the chunked exemplars shown in training provided the same correct chunk dependency information that was available in the diagram. The results showed that viewing the exemplars within the context of the diagram at training led to greater accuracy at test. This main effect of display content was qualified by a content by type interaction where the diagram produced greater accuracy than the chunks only when animated. This means that the utility of the diagram is not just in providing information about correct chunk dependencies. If that were the case, participants who viewed the chunks should have been as accurate. At least when animated, the diagram provides something more than just information on chunk dependencies.

A second hypothesis was that the utility of the animated diagram in Sun and Mathews (2004) may have been that it encouraged deeper processing (Craik & Lockhart, 1972) because participants were required to make predictions about which letter would appear next in the diagram, rather than making one-to-one comparisons as in the other conditions. If that were the case, participants who made predictions in the current study should have been more accurate and faster than those who did not make predictions. The results of the current study show no effect of prediction on speed or accuracy on the cued-generation test.

The third hypothesis was that the temporal element added by animation in Sun and Mathews (2004) may have made the dependencies between letters more salient. If this were the case, participants in the current study who viewed an animated display should have been more accurate on the cued-generation test than those who viewed a static display. There was no main effect of display type on accuracy during the cued-generation test suggesting that animation by itself did not facilitate learning. Further evidence to reject this hypothesis is that regardless of display type (animated or static), participants who viewed chunks in training responded accurately on the grammaticality judgment test when errors were of the within–chunk type. Animation across display type did not facilitate learning. Only animation of the diagram facilitated learning.

This pattern of results suggests that the utility of the animated diagram does not lie within one factor, as the diagram only produced fast and accurate performance on the cued-generation test when it was animated. The animated diagram provided the same information about correct dependencies between the chunks as the conditions in which the exemplars were parsed into chunks. However, the diagram also provided additional information that was not available in the chunk conditions. Only the diagram, with its pathways between each state, showed participants what **cannot** come next. Participants in the chunk conditions see that the chunk "TSX" can follow "CVC", but they are not shown explicitly that "TSX" cannot follow "SCP". It appears that this information only became salient when the diagram was animated, suggesting that the temporal element provided by animation combined with information about what is and is not allowable was responsible for the accurate performance at test.

The current results suggest that the fast and accurate participants were not using an explicit model of the grammar during the cued-generation test as in Domangue et al. (2004). The pen-and-paper exemplar diagramming task in Domangue et al. encouraged participants to parse the exemplars into individual letters and place them in the diagram. Doing so allowed participants to develop an explicit model of the grammar which resulted in very accurate performance at test. The disadvantage of using the explicit model was that participants performed slowly at test.

Unlike Domangue et al. (2004), in the present training task participants were focused on rapidly perceiving whole (corrected) exemplars. The structured information (diagram or chunks) could be used to correct the target string, but the emphasis of the training task was on producing an intact whole string. Animating the diagram focused participants' attention on the information relevant at the current point in time. By forcing quick decisions, participants were encouraged to process the exemplars in chunks rather than in a letter-by-letter fashion. The animation focused attention on each portion of the model as it became relevant. The memory-based processing, developed by processing exemplars in a chunk-by-chunk fashion, combined with model-based processing, used to correct the strings resulted in fast and accurate performance on the cued-generation test. All participants were fast, because they processed the exemplars in chunks. Participants who saw the static diagram were not as accurate because they could not effectively divide their attention between the edit task and the entire model at the same time. Only participants who viewed the animated diagram were fast and accurate because they processed the exemplars in chunks and developed knowledge about correct and incorrect placement of the chunks.

In the cued-generation test, whole chunks were missing, requiring participants to insert whole chunks into their appropriate position within the letter string. Knowledge of which chunks could and could not be placed in each position was acquired most effectively through training with the animated diagram. This conclusion is also supported by the results of the grammaticality judgment test. When errors were within a chunk, participants who viewed chunks at training were more accurate than those who viewed the diagram. This is reasonable, as the format of their training encouraged better intra-chunk knowledge than those who had never seen exemplars already parsed into chunks. Mere exposure to the chunks was sufficient for intra-chunk knowledge.

The requirements of the grammaticality judgment test when the error was due to a valid chunk placed in the wrong position were similar to those in the cued-generation test. Participants needed to know which chunks could and could not be placed in each position within the string. The results were similar to the cued-generation test, such that participants who saw the diagram were more likely to identify this type of error than those who saw chunks at training. Additionally, while not significant, there was a trend for an interaction (p = 0.7) in which the advantage for the diagram is only present when it is animated, which follows the results of the cued-generation test.

The results of this study have implications for using animation as a training tool. Tversky et al. (2002) have questioned the notion that animation can facilitate learning. One major criticism is that animation often provides more information than static graphics. The present study provided the same information to all participants who saw the exemplars within the context of the diagram. However, some of the information only became salient when the temporal element provided by animation was added, demonstrating the facilitative effects of animation on learning. When information can be acquired simply through mere exposure (i.e. intra-chunk knowledge), static displays are just as effective as animated displays. However, when information is complex and demands are placed on attentional resources, animation can focus attention on the relevant portion of the display. In effect, animating a complex display can make it easier for the learner to use, as it reduces the cognitive demands placed on them by directing their attention.

The results of the grammaticality judgment test further show that knowledge of chunks (Servan-Schreiber & Anderson, 1990) is not sufficient for accurate performance on this task. When errors were within a chunk, participants who viewed chunks at training were accurate at identifying those errors. However, when the error was due to a valid chunk placed in the wrong position within the exemplar, those same participants were no more accurate than the no-training control.

REFERENCES

Anderson, J. R. (1976). *Language, Memory, and Thought.* Oxford: Erlbaum.

Caramazza, A., McCloskey, M. & Green, B. (1981). Naïve beliefs in "sophisticated" subjects: Misconceptions about trajectories of objects. *Cognition, 9*, 117-123.

ChanLin, L. J. (1998). Animation to teach students of different knowledge levels. *Journal of Instructional Psychology, 25*, 166-175.

Craik, F. I. M., & Lockhard, R. S. (1972). Levels of processing: A framework for memory research. *Journal of Verbal Learning and Verbal Behavior, 11*, 671-684.

Domangue, T. J., Mathews, R. C., Sun, R., Roussel, L. G., & Guidry, C. E. (2004). Effects of model-based and memory-based processing on speed and accuracy of grammar string generation. *Journal of Experimental psychology: Learning, Memory, 30,* 1002-1011.

Greenberg, M. S., Westcott, D. R., & Bailey, S. E. (1998). When believing is seeing: The effect of scripts on eyewitness memory. *Law & Human Behavior, 22(6),* 685-694.

Large, A., Beheshti, J., Breuleux, A., & Renaud, A. (1996). The effect of animation in enhancing descriptive and procedural texts in a multimedia learning environment. *Journal of the American Society for Information Science*, *47*, 437-448.

Mathews, R. C. (1991). The forgetting algorithm: How fragmentary knowledge of exemplars can abstract knowledge. *Journal of Experimental Psychology: General, 120,* 117-119.

Mathews, R. C., Buss, R. R., Stanley, W. B., Blanchard-Fields, F., Cho, J. R., & Druhan, B. (1989). Role of implicit and explicit processes in learning from examples: A synergistic effect. *Journal of Experimental psychology: Learning, Memory, and Cognition, 15,* 1083-1100.

Mathews, R. C., & Cochran, B. P. (1997). Project grammarama revisited: Genereativity of implicitly acquired knowledge. In M. A. Stadler & P. A. Frensch (eds.), *Handbook of implicit knowledge* (pp. 223-259). Thousand Oaks, Ca: Sage.

Mathews, R. C. & Roussel, L. G. (1997) Abstractness of implicit knowledge: A cognitive evolutionary perspective. In D. Berry (Ed.). *How Implicit is Implicit Learning?* New York: Oxford University Press.

Miller, G. A. (1958). Free recall of redundant strings of letters. *Journal of Experimental Psychology, 56,* 485-491.

Norman, D. A. (1993). *Things that make us smart: Defining human attributes in the age of the machine.* Reading, MA: Addison-Wesley.

Palmiter, S. L., Elkerton, J., & Baggett, P. (1991). Animated demonstrations vs. written instructions for learning procedural tasks: A preliminary investigation. *International Journal of Man-Machine Studies, 34,* 687-701.

Perruchet, P., & Pacteau, C. (1990). Synthetic grammar learning: Implicit rule abstraction or explicit fragmentary knowledge? *Journal of Experimental Psychology: General, 119,* 264-275.

Perruchet, P., Vinter, A., Pacteau, C., & Gallego, J. (1992). The formation of structurally relevant units in artificial grammar learning. *The Quarterly Journal of Experimental Psychology, 55A,* 485-503.

Pothos, E. M. (2005). Expectations about stimulus structure in implicit learning. *Memory and Cognition, 33,*177-181.

Reber, A. S., (1969). Transfer of syntactic structure in synthetic languages. *Journal of Experimental Psychology, 81,* 115-119.

Rieber, L. P., Boyce, M. J., & Assad, C. (1990). The effects of computer animation on adult learning and retrieval tasks. *Journal of Computer-Based Instruction, 17,* 46-52.

Schnotz, W. & Grzondziel, H. (1990). Individual and co-operative learning with interactive animated pictures. *European Journal of Psychology of Education, 14,* 245-265.

Servan-Schreiber, E., & Anderson, J. R. (1990). Learning artificial grammars with competitive chunking. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 16,* 592-608.

Shanks, D. R., & Cannon, S. (2002). Effects of a secondary task on “implicit” sequence learning: learning or performance? *Psychological Research, 66,* 99-109.

Sun, R. & Mathews R. C. (2004). Exploring the interaction of implicit and explicit processes to facilitate individual skill learning. Unpublished manuscript.

Tversky, B., Morrison, J. B., & Betrancourt, M. (2002). Animation: Can it facilitate? *Internation Journal of Human-Computer Studies, 57,* 247-262.

APPENDIX A

REPRESENTATIVE EXEMPLARS FROM DOMANGUE ET AL. (2004)

Set A	Set B
CVCPTTTTTVV	CVCPTVV
CVCPVPS	CVCTSXXTVPS
CVCTSXXVV	CXPTTVV
CXPTVPXVPS	CXPVPS
CXTSSSXXVV	CXTSSXXVV
CXTSXXVV	CXTXS
CXTXXTTTTVV	SCPTTTVPXVV
SCPTTVPXTVV	SCPTVPXVV
SCPVPXTTTVV	SCTSSSXXVV
SCTSSXXTTVV	SCTSXXTTVV
SCTSXXTTVPS	SCTXXVV

APPENDIX B

LIST OF CHUNKS

Parentheses denote letter of chunk that can be repeated.

CVC
CXP
CXT
SCP
SCT
T(S)
P(T)
(S)
(T)
TSX
VPS
(VPX)
VV
XX
XS
TXX
P

APPENDIX C

COMPLETE LIST OF INVALID LETTER STRINGS FROM GRAMMATICALITY JUDGMENT TEST

Interchunk Errors

CXPPTVPXTVV	SCTSSSXXTXS	CXPVPSVV	CVCVPXPTVV
CVCXXTVPS	CXTVV	CVCCVCVV	CVCXXTSVPS
CVCSSXXVPS	CXPTTTTTTSX	SCPSCTVV	CVCVPXPTTVV
CXPSSXXTVV	CXTSXXVPXXS	SCPTCXPVPS	CVCXXTSTVPS
CXTTVV	CVCPTTVPXXS	CXTSSXSTTVV	CXTSTTXXVV
SCTTVPXTVV	SCTSXXTXS	SCPSCTTTVV	CVCVPXPVV
SCTPTTTTVV	CXPVPXTTXS	CVCTXSVPS	CXTVPXXXVV
SCPXXVPS	CVCPTTTTTXS	SCTCXTTTVV	CVCXXTSSSVV
CXTPTTTTVPS	CXPTVPXXS	SCTVPSXS	SCTSSTTXXVV
SCTTVPS	CVCTSSXXTSX	CVCTXSXS	CVCTTTTTXS
CXPTTTXXVV	PTTTTVPXTVV	CXPXX	CXTTTTTTTXS
SCTSVPXTTVV	VPSTSXXVV	SCPSS	SCTTTTTTTXS
CVCTXXPTVPS	VPXTXXTTTVV	CVCTSXXTTPT	CXTTTTTTXS
SCTVVTVV	XXVPXVPS	CVCPTVPXSCT	CVCTTTTXS
CXTXXTPTTVV	VVXXTTTVPS	CXTSSSSXXVV	CXTXXSSSSVV
SCTVVTTTVPS	TSXVPXTTVV	SCPTTTCVC	CXTXXSSVPS
SCTSSPTTVV	PVVTXS	SCTXXTTCXP	CXPSSVV
CXTSSPTVPS	VPSSSXS	CXTSSSXXVPX	CVCPSSSVV
SCPPTTVPXVV	SSSTVPXTVPS	SCPTVPXTXX	CXPSSSSVPS
CXPSSSVPXVV	XSTVPXVPS	CXTSTTXXVPS	

Intrachunk Errors

SVSSXXTVPS	CXTSSSCVTVV	SCSTTVV	SCPTVTXTTVV
CSTVPXTVV	SCTSTCTTVPS	CXCTSSSSSXS	CXTPXTTTVV
STSTSSXXTVV	CVCXTPXS	CVXPVPXTVPS	CVCTSSVXVV
CTCTTTTTVV	CXPTSCSTVV	CSTSXXTVV	CXTSSSSTS
SPCTTTVPS	SCTSXXPTCVV	SCXVPXTTTVV	SCTSSXXTVSS
CPSTXXTVV	SCTCS	SVTXXVV	CXPTTTTTPV
STVSSXS	SCPTTVPXTPS	CXXSXS	CXTSSXXTV
STCVPXTTTVV	CXPTTVPXSV	SXPTTTTVPS	SCPTTTTTVVS
CCXXXVPXVPS	CVCPTCV	CVSTXXTTVV	SCTSTS
SVTVPS	SCPTTXSS	CXPTVTXTVPS	SCPTTTXPS
CXVSXXTVPS	CVCTSXXTCV	CXTPXTTVV	SCPVTS
CXTSTCVPS	CXTXXTPV	CVCTXPVV	CVCTXXTSPS
CXTSSSCTVV	CXTXXCS	CXPVPCTVV	SCTXXTTTTSV
SCTVTVXXVV	CXPVPXVXS	CVCTXXVPVVV	
SCPPTSVPXVV	CXTSXXCV	CVCVTTVPS	
CXPTSTVVV	CSCPVPXVPS	SCTSTXTTTVV	

VITA

Bill Sallas received his Bachelor of Science in psychology and his Master of Education from the University of Illinois, Urbana-Champaign. He taught for two years in the Chicago Public School system. Bill is currently a graduate student at Louisiana State University, where he studies human learning. His interests also include using technology to facilitate learning and on-the-job performance.